Abstract

eTrice provides an excellent implementation of the ROOM modeling language, equipped with graphical (Graphiti) as well as textual (XText) editors for both structural and behavioral parts. ROOM (Real Time Object Oriented Modeling) is a domain specific language for distributed, event driven, real time systems and provides the clearest, simplest and the most complete modeling concepts for the real time domain. Through **Extended Model Validation and State Graph Proposal Generation Based on Protocol Semantics,** the idea is to equip the protocol (set of incoming and outgoing messages for each port of the actor) with a semantic description defining the allowed message replies for each incoming message. This idea is very powerful as it would be able to check for incomplete and invalid state diagrams and generate proposals for state machine. More features and advantages of this idea are explained in detail in the following sections.

Detailed Information

The basic elements of ROOM are the actors with their ports and protocols. The protocol provides a formal interface description. The port is an interaction point where the actor interacts with its outside world. Each port has exactly one protocol attached. The sum of all ports builds up the complete interface of an actor. Each port can receive messages, with or without data, which are defined in the attached protocol. Each message will be handled by the actors behavior (state machine) or will be delegated to the actors internal structure. Currently, there is no tool which the user can use to verify whether his state diagram is complete and valid, unless he tries and execute the model. Depending on the error, it can be quiet difficult to identify the root cause which invalidates the model. To prevent this, a protocol can be defined with a semantic description consisting of the possible allowed message replies to each incoming message. Then eTrice would be able to generate proposals for the possible valid reply messages to incoming messages and also at the same time check, whether the current state diagram is valid and complete by using methods of abstract model execution. In the precursor tool, Trice, it has already been shown that this idea is very powerful. However, the algorithm used in the precursor tool is not quite exact, and thus we hope to gain a deeper understanding of the algorithm and to improve it . For the developers and users of ROOM and eTrice, the following will be the key benefits:

1) Extended Model validation would enable the users and developers to rectify their mistakes while designing the state machine on the go and prevent them from making mistakes. Unhandled messages and messages that should be sent will be detected. As a side product, unconnected parts of the state graph are found as well.
2) The validation engine will produce warning markers and quick fixes in the textual model.
3) Through State Graph Proposal generation in the graphical editor, it would become relatively easy to design state machines for complex software systems, as eTrice would generate valid proposals for each incoming and outgoing messages through proper annotations.
4) The user would have to worry less about the problem of incomplete state machines as any incompleteness will be highlighted through proper graphical notation in eTrice.
5) Provide an easier interface for adding the states and transitions to state machine by clicking or dragging on the generated proposals.

Required Resources

The project would require eTrice sources, XText 2.1, Graphiti 0.8.0 and familiarity with the grammar language of the ROOM implementation. Also, eclipse platform would be needed for plugin development. We would also use the good features from the former implementation of this feature in the precursor tool Trice.

Background

I have already setup the eTrice sources along with XText and Graphiti and downloaded the source code from github repository. I hope to familiarize myself better with grammar language of the ROOM implementation and also with the corresponding EMF model. I have also already done all the relevant eTrice tutorials . I am also familiarizing myself with as much of the source code as possible, so I'll have to spend minimal time on it when the program starts and can focus directly on the development of the project. I am also familiar with basic eclipse plug-in development, and will learn more about it before the summer to make myself proficient in the same.

Deliverables

The key deliverable would be the same as expected benefits from the project. However, for the sake of clarity they are listed here :

1) Provide user with an option to equip protocol with a semantic description of allowed reply messages for each incoming and outgoing message.
2) Extended Model Validation : Develop an algorithm to verify whether the current state machine is complete and valid based on the protocol semantics defined by the user.
3) Generate markers and graphical proposals for the allowed transitions and states based on the semantic description in the graphical editor.
4) Implement the interface for clicking and dragging a proposal to create the state and transition in the state machine .

Schedule

After detailed discussion with Henrik, we have identified the following three milestones for the project :

1) Understand and extend the way XText deals with markers and quick fixes. Define and introduce a new extension point and define and implement the FSM executor (or abstract state machine execution algorithm).
2) Implement the Connectivity and Protocol Validator using the executor
3) Use the new validator in the graphical FSM editor.

Even though this is fair amount of work, I am confident that we'll be able to complete it. From my side, 50-60 hours of work per week is guaranteed as this is my only commitment for the entire duration of the summer and I will treat like any other full time internship with utmost seriousness.

Expectations

I hope to develop a good working equation with my mentor. I'll try and work through the project under the guidance of my mentor and consult him in case of ambiguity or roadblock (only when I have tried my best to work through the problem myself) . This would ensure maximum learning on my part and would also ensure that I am able to use my innovation and creativity to give my best to the project. Any extra learning resources that I might need, I'll consult with my mentor for the best possible source so I have to waste minimal time on finding the best resource to read up. I hope to become proficient with Eclipse plug in development and eTrice after this project.

Contact Information

My contact information is as follows :

Name : Rohit Agrawal

Institute : Nanyang Technological University, Singapore

Where I will be in Summer : Lucknow , India

Phone : +65 93232726 , +91 9044145746

I am a third year undergraduate student at Nanyang Technological University , majoring in Electrical and Electronic Engineering and specializing in Intelligent System Design and Control Engineering. I am also pursuing a Minor in Computing. Though this is my first time, I am very interested in taking part in Google Summer of Code and contributing to the Eclipse Open source community. I also hope to be able to contribute even after the program is over.

Prior to this , I have done an internship in Bank of America Merrill Lynch , Singapore as a QA Analyst in the Electronic Trading Technology team. I am familiar with many programming languages like Java, C++, Perl, Python, Processing etc. I am proficient in usage of most of the data structures and good at algorithms . I am familiar in working on both Windows and Linux operating systems.